



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/781,307	02/18/2004	John G. Beltran	G&C 30566.296-US-U1	4847
55895 7590 01/24/2008 GATES & COOPER LLP HOWARD HUGHES CENTER 6701 CENTER DRIVE WEST, SUITE 1050 LOS ANGELES, CA 90045			EXAMINER ABDUL-ALI, OMAR R	
			ART UNIT 2178	PAPER NUMBER
			MAIL DATE 01/24/2008	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/781,307

Applicant(s)

BELTRAN ET AL.

Examiner

Omar Abdul-Ali

Art Unit

2178

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 11/07/2008.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-18 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-18 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- ☐ Notice of References Cited (PTO-892)
- ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____
- ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
- ☐ Notice of Informal Patent Application
- ☐ Other: _____

DETAILED ACTION

The following action is in response to the response filed November 7, 2007. Amended Claims 1-18 are pending and have been considered below.

1. Examiner's Note: The 35 U.S.C. 101 rejections have been withdrawn as necessitated by Applicant's amendments.

Specification

2. The use of the trademarks Netscape Navigator™ and Microsoft Internet Explorer™ has been noted in this application. It should be capitalized wherever it appears and be accompanied by the generic terminology. The use of all capital letters to distinguish the trademarks is insufficient when the trademark symbol ™ does not accompany these trademarks.

Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner which might adversely affect their validity as trademarks.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the

subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains.

Patentability shall not be negated by the manner in which the invention was made.

7. Claims 1-18 remain rejected under 35 U.S.C. 103(a) as being unpatentable over Hirsch (US 6,915,301).

Claim 1: Hirsch discloses a method and computer system for dynamic properties of software objects comprising:

a. receiving a reference to an object instance having a dynamic property that is created at runtime for the object instance on a per-instance basis and is not stored with the object (column 11, lines 37-58);

Hirsch discloses retrieving reference to a property source instance (data source) from an association between the object and the property source instance (binding), wherein the property source instance creates and supplies to dynamic property (column 4, lines 11-34/column 11, lines 37-59), but does not explicitly disclose creating and supplying an initial value for the dynamic property for/to the object instance. However, Hirsch does disclose data sources generate values in a scene, and objects and their properties are bound to data sources in each scene. Furthermore, providing initial values was a well-known technique in the art at the time the invention was made.

Therefore it would have been obvious to one having ordinary skill in the art at the time

the invention was made that the data sources may supply an initial value for the dynamic property for/to the object instances in Hirsch. One would have been motivated to create and supply an initial value for the dynamic property in order to initialize the property to an initial value at runtime.

c. providing the reference to the object and the reference to the property source instance to a control which is configured to: (i) retrieve the dynamic property from the property source and (ii) display the property (object inspector window) in a user interface (column 11, 37-67).

Claim 2: Hirsch discloses a method and computer system for dynamic properties of software objects as in Claim 1 above, further comprising:

a. the dynamic property is provided by an application that is extending an object property set of the object (column 12, lines 41-58).

Claim 3: Hirsch discloses a method and computer system for dynamic properties of software objects as in Claim 1 above, further comprising:

a. reference to the property source instance is retrieved from a mapping (binding) of property source instances to object class (column 3, lines 8-27/column 12, lines 41-58).

Claim 4: Hirsch discloses a method and computer system for dynamic properties of software objects as in Claim 1 above, further comprising:

a. the control is further configured to: retrieve the standard properties for the object; and display the standard properties (column 11, lines 52-67).

Claim 5: Hirsch discloses a method and computer system for dynamic properties of software objects comprising:

a. an object instance having a dynamic property that is created at runtime for the object instance on a per-instance basis and is not stored with the object (column 11, lines 37-59);

Hirsch discloses a property source instance wherein the property source instance creates and supplies the dynamic property (column 11, lines 37-59) but does not explicitly disclose creating and supplying an initial value for the dynamic property for/to the object instance. However, Hirsch does disclose data sources generate values in a scene, and objects and their properties are bound to data sources in each scene (column 4, lines 11-34/column 11, lines 37-59). Furthermore, providing initial values was a well-known technique in the art at the time the invention was made. Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made that the data sources may supply an initial value for the dynamic property for/to the object instances in Hirsch. One would have been motivated to create and supply an initial value for the dynamic property in order to initialize the property to an initial value at runtime.

c. an association between the object and the property source instance (column 3, lines 8-27);

d. a host configured to: (i) retrieve a reference to the object instance; (ii) retrieve a reference to the property source; (iii) provide the reference to the object and the reference to the property source to a control which is configured to (1) retrieve the dynamic property from the property source and (2) display the property in a user interface (column 11, lines 37-67).

Claim 6: Hirsch discloses a method and computer system for dynamic properties of software objects as in Claim 5 above, further comprising:

a. the dynamic property is provided by an application that is extending an object property set of the object (column 12, lines 41-58).

Claim 7: Hirsch discloses a method and computer system for dynamic properties of software objects as in Claim 5 above, further comprising:

a. a mapping of property source instances to object classes, wherein the host is configured to retrieve the reference to the property source instance from the mapping (column 12, lines 41-58).

Claim 8: Hirsch discloses a method and computer system for dynamic properties of software objects as in Claim 5 above, further comprising:

a. the control is further configured to: retrieve the standard properties for the object; and display the standard properties (column 11, lines 52-67).

Claim 9: Hirsch discloses a method and computer system for dynamic properties of software objects comprising receiving a first object having a first property wherein the first object provides a control (calendar control) that defines a first user interface for displaying and editing the first property, but does not explicitly disclose the control is an ActiveX control. However, Hirsch does disclose supporting ActiveX controls (column 12, lines 8-31). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to provide an ActiveX control to display and edit the first property. One would have been motivated to provide an ActiveX control to display and edit the first property in order to make the design more efficient.

- b. creating a list of one or more object properties to be displayed, wherein the list includes the first property (column 12, lines 8-31);
- c. instantiating the custom ActiveX control (column 12, lines 8-31);
- d. displaying the object properties in the list, wherein the display of the first property comprises the first user interface defined by the instantiated ActiveX control, wherein the property may be edited through the first user interface (column 12, lines 8-31).

Claim 10: Hirsch discloses a method and computer system for dynamic properties of software objects as in Claim 9 above, further comprising:

- a. instantiating one or more stock ActiveX controls that define one or more additional user interfaces for displaying and editing remaining object properties in the

list, wherein the stock ActiveX controls are not provided by any object containing one or more of the remaining object properties (column 12, lines 8-31).

Claim 11: Hirsch discloses a method and computer system for dynamic properties of software objects as in Claim 10 above, further comprising:

a. the first user interface and additional user interfaces are displayed in a single dialog box (column 12, lines 8-31).

Claim 12: Hirsch discloses a method and computer system for dynamic properties of software objects as in Claim 9 above, further comprising:

a. an application programming interface provides the ability to push the first object to a second object for display (column 12, lines 41-58).

Claim 13: Hirsch discloses a method and computer system for dynamic properties of software objects comprising:

a. one or more objects, wherein each object has one or more object properties (column 11, lines 37-59);

b. a property inspector configured to (i) interrogate the one or more objects to discover one or more object properties to be displayed; (ii) create a list of the one or more object properties to be displayed; (iii) instantiate and host one or more property editors (column 11, lines 37-59/column 12, lines 1-21);

Hirsch discloses one or more property editors wherein: (i) one of the property editors comprises a custom control specified by one of the objects, but does not explicitly disclose the control is an ActiveX control. However, Hirsch does disclose supporting ActiveX controls (column 12, lines 8-31). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to provide an ActiveX control specified by one of the objects. One would have been motivated to provide an ActiveX control specified by one of the objects in order to make the design more efficient.

Hirsch discloses (ii) the custom ActiveX control defines a custom graphical user interface for displaying and editing one of the object properties (column 12, lines 8-31).

Claim 14: Hirsch discloses a method and computer system for dynamic properties of software objects as in Claim 13 above, further comprising:

a. one of the property editors is comprised of a stock ActiveX control that defines an additional user interface for displaying and editing one or more additional properties in the list, wherein the stock ActiveX control is not provided by one of the objects that contains the one or more additional properties (column 12, lines 8-31).

Claim 15: Hirsch discloses a method and computer system for dynamic properties of software objects as in Claim 14 above, further comprising:

a. the custom graphical user interface and additional user interfaces are displayed in a single dialog box (column 11, lines 37-59/column 12, lines 1-21).

Claim 16: Hirsch discloses a method and computer system for dynamic properties of software objects as in Claim 13 above, further comprising:

a. an application programming interface configured to provide the ability to push the one or more objects to the property inspector for display (column 12, lines 41-58).

Claim 17: Hirsch discloses a method and computer system for dynamic properties of software objects comprising:

a. an object instance of a class, wherein (i) an initial value for one or more static properties of the class are assigned at run time; and (ii) the object instance has a dynamic property that is generated by a property source instance at runtime for the object instance on a per-instance basis and are not stored with the object (column 12, lines 20-58/column 11, lines 37-59). Hirsch does not explicitly disclose an initial value is generated and supplied by a property source instance. However, Hirsch does disclose data sources generate values in a scene, and objects and their properties are bound to data sources in each scene. Furthermore, providing initial values was a well-known technique in the art at the time the invention was made. Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made that the data sources may supply an initial value for the dynamic property for/to the object instances in Hirsch. One would have been motivated to create and supply an initial value for the dynamic property in order to initialize the property to an initial value at runtime.

b. an association between either: (i) the object instance and the property source instance; (ii) the class and the property source instance (column 11, lines 37-61)

c. a user interface component that displays a collection of properties of the object instance including the one or more static properties and the dynamic property on a display device(column 11, lines 37-61), wherein the user interface component is configured to:

(i) retrieve a reference to the object instance (column 11, lines 49-63);

(ii) retrieve the one or more static properties from the object instance (column 11, lines 49-63);

(iii) access the association to determine the property source instance associated with the object instance (column 11, lines 37-61);

(iv) call a method of the determined property source instance with the reference to the associated object instance (column 11, lines 37-61);

(v) receive the dynamic property, from the property source instance, wherein the property source instance dynamically generated the dynamic property (column 11, lines 49-63). Hirsch does not explicitly disclose an initial value is generated and supplied by a property source instance. However, Hirsch does disclose data sources generate values in a scene, and objects and their properties are bound to data sources in each scene. Furthermore, providing initial values was a well-known technique in the art at the time the invention was made. Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made that the data sources may supply an initial value for the dynamic property for/to the object

instances in Hirsch. One would have been motivated to create and supply an initial value for the dynamic property in order to initialize the property to an initial value at runtime.

(vi) display the static property and the dynamic property on the display device (column 11, lines 49-63).

Claim 18: Hirsch discloses a method and computer system for dynamic properties of software objects as in Claim 9 above, further comprising custom controls are provided on a per-instance basis, but does not explicitly disclose the controls are ActiveX controls. However, Hirsch does disclose supporting ActiveX controls (column 12, lines 8-31). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to provide an ActiveX control on a per-property basis. One would have been motivated to provide an ActiveX control on a per-property basis in order to make the design more efficient.

- b. two or more object properties are in the properties list (column 12, lines 8-31);
- c. the two or more object properties are displayed in a list, wherein each of the two or more object properties are displayed using user interfaces defined by the custom ActiveX controls (column 12, lines 8-31).

It is noted that any citation [[s]] to specific, pages, columns, lines, or figures in the prior art references and any interpretation of the references should not be considered to be limiting in any way. A reference is relevant for all it contains and may be relied upon

for all that it would have reasonably suggested to one having ordinary skill in the art.

[[See, MPEP 2123]]

5. Applicant's arguments filed November 7, 2007 have been fully considered but they are not persuasive.

Claims 1, 5, and 17: Applicant argues, "Hirsch [does not explicitly disclose] the creation of a property source instance for a property of a separate object instance." It is respectfully submitted that the disclosure of Claim 1 does not include this limitation.

Applicant argues, "Hirsch [does not explicitly disclose] the creation of a value for such a dynamic object." It is respectfully submitted that Hirsch creates and sets values at runtime based on calculated results of expressions. The creation of a value based on calculated expressions is viewed as dynamic because the value is created at runtime. Applicant argues Hirsch's dynamic properties already exist for an object but are merely evaluated at runtime, however, the creation of a value at runtime based on a calculated expression is dynamic.

Applicant argues, "Nowhere in Hirsch is there even a remote reference to a control receiving both a reference to an object instance and a property source instance, and then retrieving the dynamic property from the property source instance and displaying such a property." It is respectfully submitted that Hirsch discloses the limitation as claimed above. Specifically, Hirsch discloses an object inspector window displays the object, the object's properties (static and dynamic), and receives the object

properties from a data source. The retrieval of this information by the object inspector window to display the information provides a reference between the property source and the object instances.

Claims 9 and 13: Applicant argues, "Nowhere in Hirsch is there any description that the object itself provides the custom control to display the properties of the object." It is respectfully submitted that Hirsch discloses the limitation as claimed above. The objects in Hirsch are selected in an object inspector window, and provide custom controls such as calendar controls when specific objects are inspected. An object that does not contain a date or time property would not provide a calendar control, giving the objects the ability to provide custom controls based on which object is being inspected.

Applicant argues, "Hirsch does not even remotely describe that the control is a custom control." It is respectfully submitted that the controls provided in Hirsch are custom controls due to the fact that different controls are used for editing particular properties of an object.

Conclusion

6. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not

mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Omar Abdul-Ali whose telephone number is 571-270-1694. The examiner can normally be reached on Mon-Fri(Alternate Fridays Off) 8:30 - 6:00 EST.

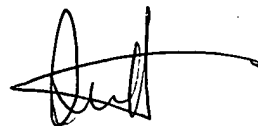
If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Stephen Hong can be reached on 571-272-4124. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Application/Control Number:
10/781,307
Art Unit: 2178

OAA
1/22/2008

Page 16

A handwritten signature in black ink, appearing to read 'Stephen Hong', with a long horizontal stroke extending to the right.

STEPHEN HONG
SUPERVISORY PATENT EXAMINER